Подводные камни высокодоступных инсталляций OLTP СУБД в облачном окружении

Андрей Бородин
Дмитрий Смаль





Яндекс





Подводные камни высокодоступных инсталляций OLTP СУБД в облачном окружении

Андрей Бородин, руководитель подразделения разработки РСУБД с открытым кодом
Дмитрий Смаль, руководитель подразделения Managed MySQL и SQL Server

- Managed Service for PostgreSQL
- Managed Service for MySQL
- Managed Service for MongoDB
- Many more DBs









Yandex.Cloud Data Platform

PostgreSQL в Яндексе

Яндекс.Почта

- Сколько-то сотен миллионов пользователей
- 1+ триллион строк, 1+ миллион запросов в секунду

Яндекс.Облако

- Несколько петабайт Постгреса
- 3+ миллиона запросов в секунду

MySQL в Яндексе

- Яндекс.Директ
- Managed MySQL
  - 400+ TB (апрель 2021)

PGConf.Russia 2021

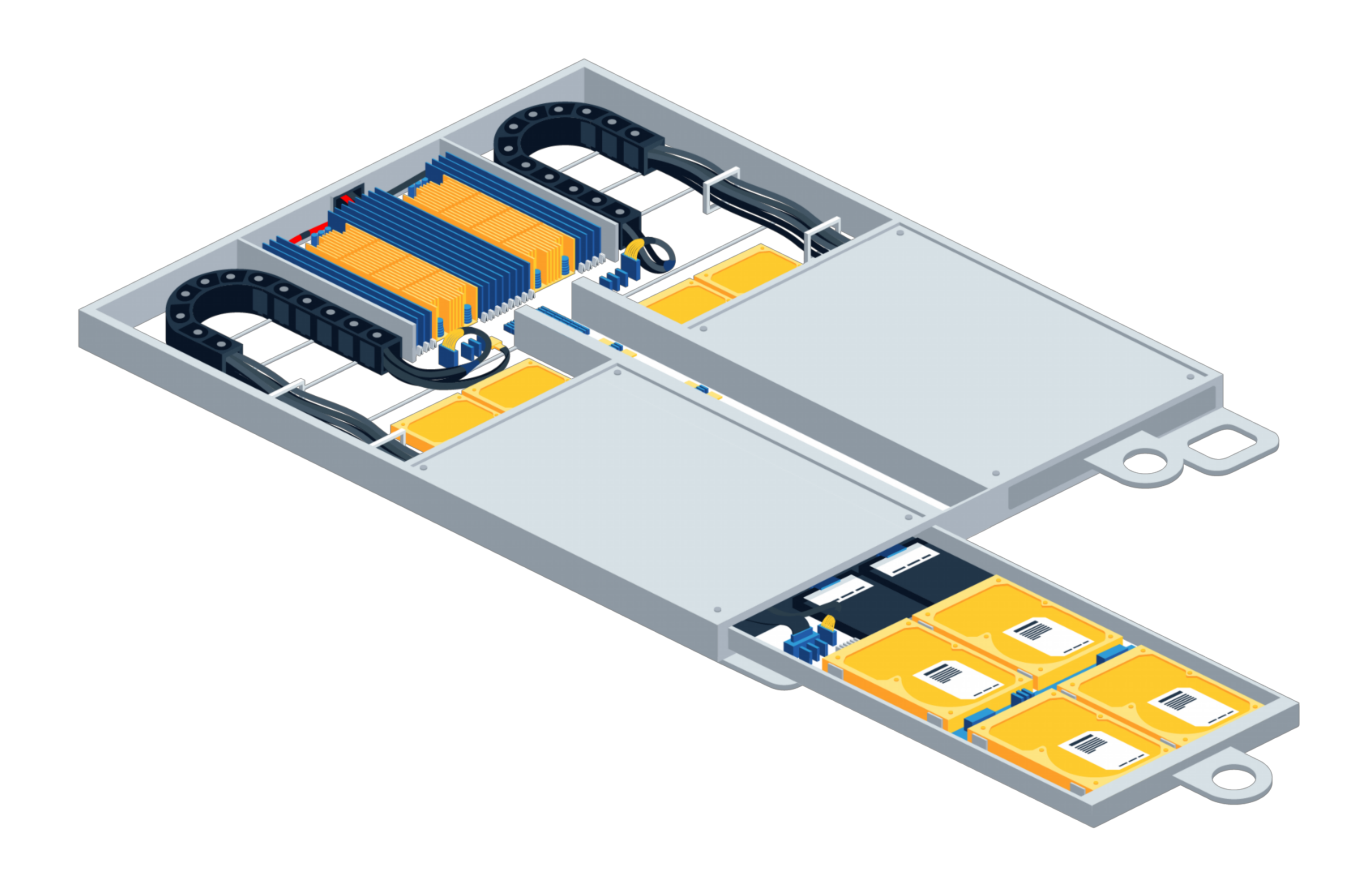
- Эксплуатация высокодоступных РСУБД с открытым исходным кодом в облачном окружении


https://pgconf.ru/2021/281357

https://clck.ru/UjAjt

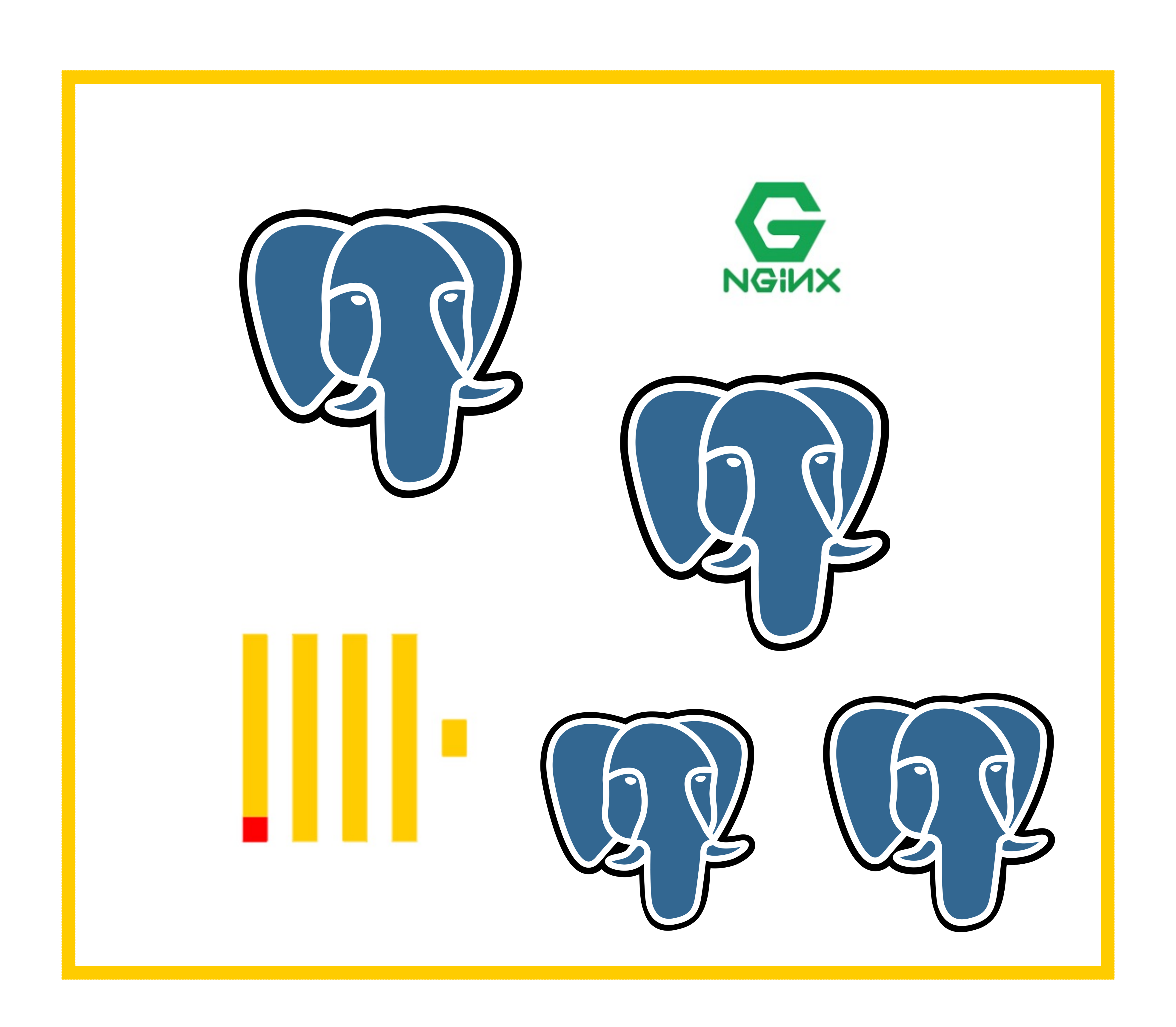
Что нужно

- Доступность 0.9999 в месяц на чтение
- Доступность 0.9995 в месяц на запись
- Масштабируемость в нескольких зонах доступности
- Актуальная копия данных в аналитической системе

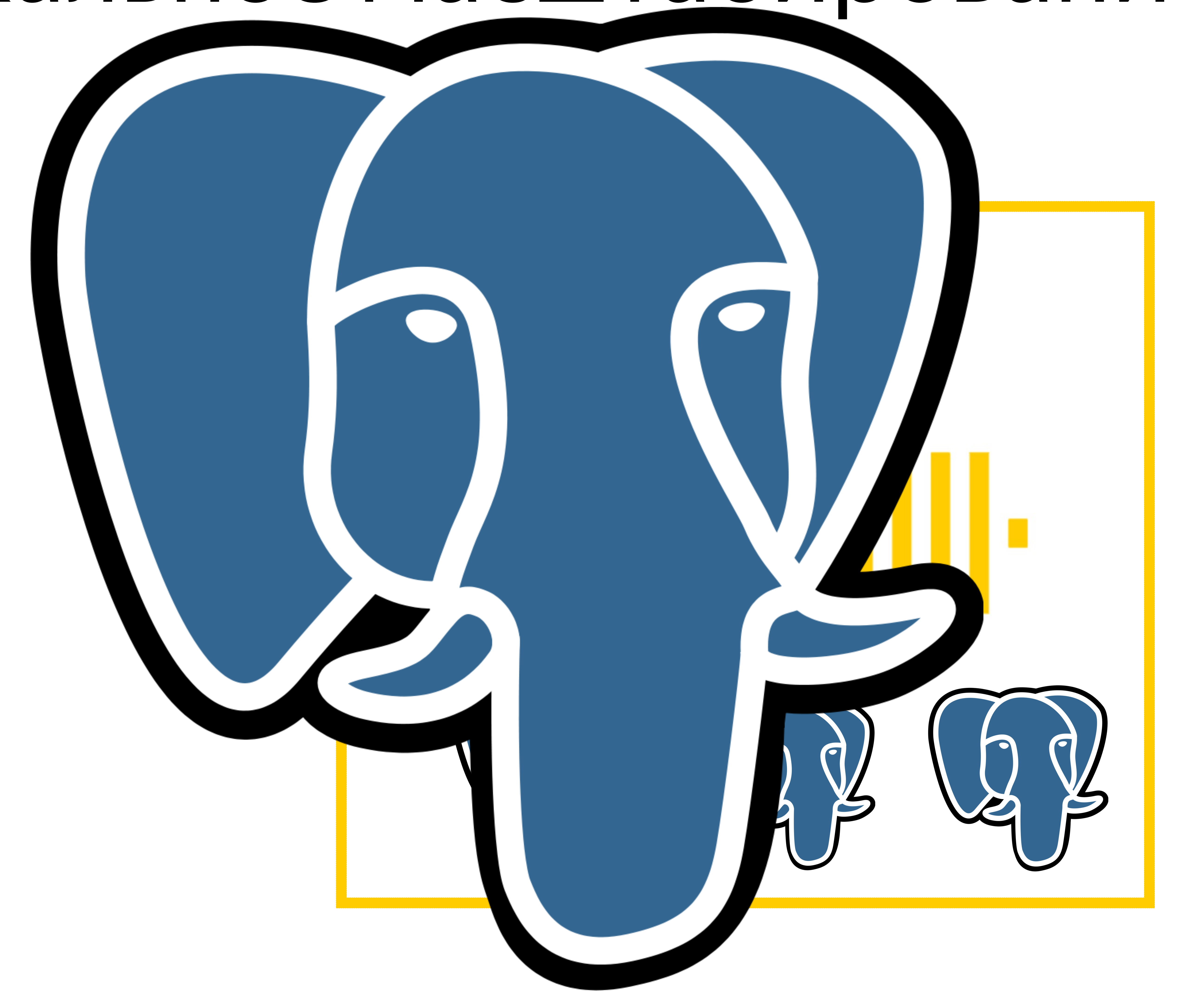




Виртуализация



Вертикальное масштабирование



Избыточность

Топология кластера в Облаке

RW Queries

WAL

Yandex Object Storage

AWS S3 API

Logical replication

OLAP\MR

Backup

Sync/Quorum replication

Async replication

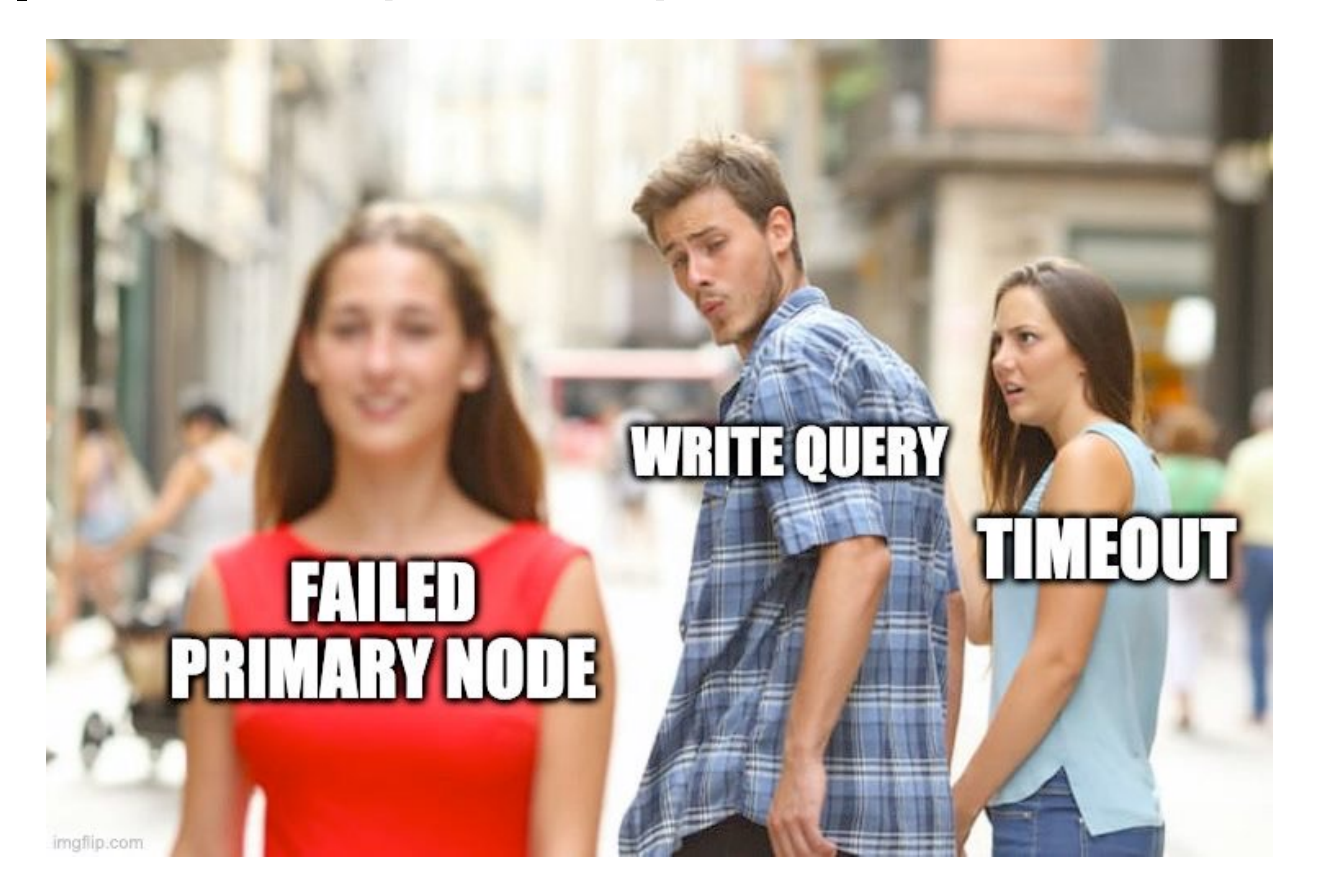
RO Queries

Как клиенту найти, куда слать пишущие запросы?

```
psql "host=<host 1 FQDN>,<host 2 FQDN>,<host 3 FQDN> \
      port=6432 \
      sslmode=verify-full \
      dbname=<DB name> \
      user= \
      target_session_attrs=read-write"
```

https://cloud.yandex.com/docs/managed-postgresql/operations/connect

Пишущие запросы при частичном отказе



Timeout'ы – это важно

tcp_user_timeout

В libpq есть бесконечные ожидания, где оно полагается на keepalives операционной системы

keepalives_count, keepalives_interval, keepalives_idle

Чтение со Standby



Проблемы

- Возможен каскадный отказ
- Не консистентные данные (как из прошлого, так и из будущего)
- Конфликт репликации
- Плохо масштабирующийся KnownAssignedXids

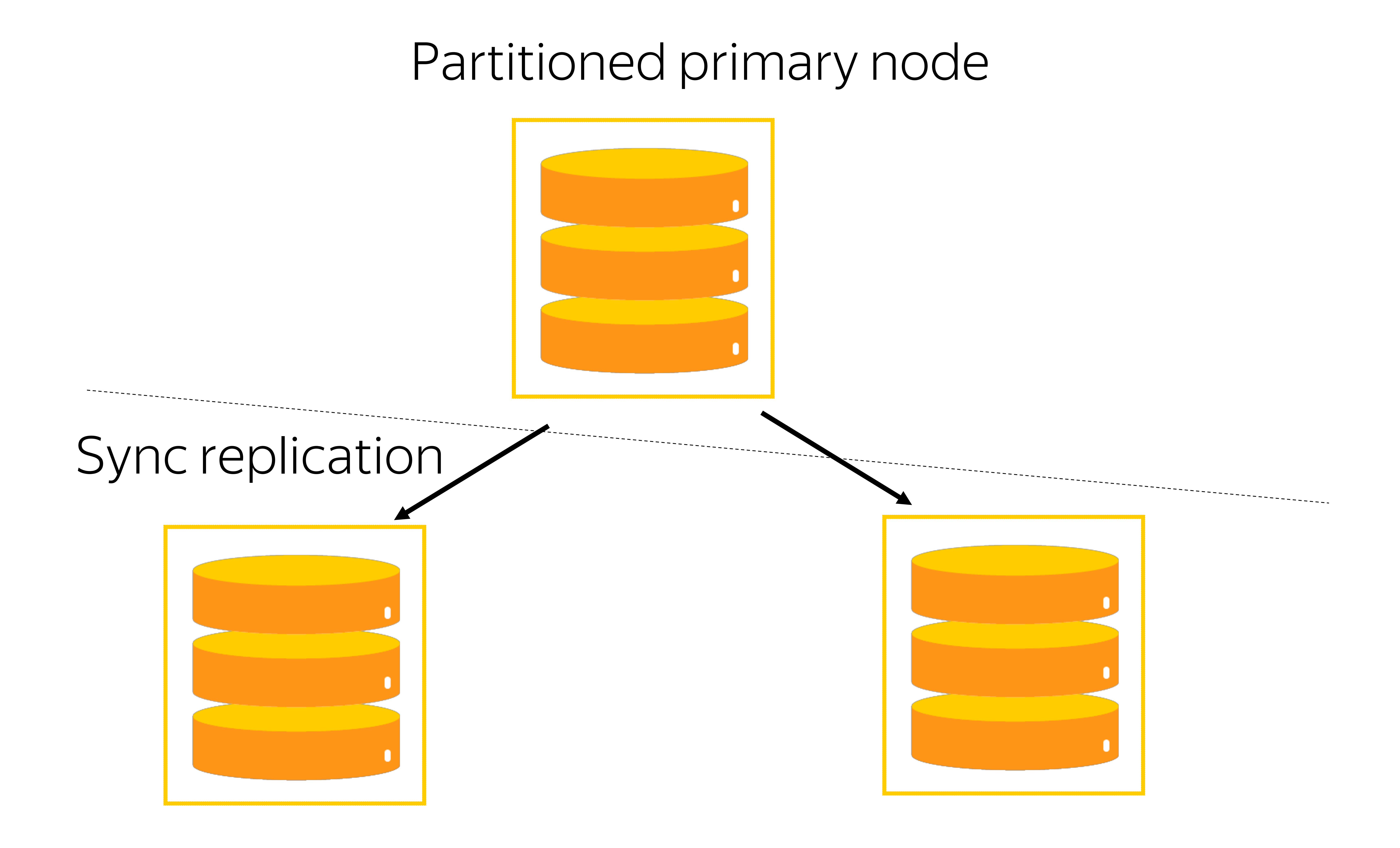
File Edit View Search Terminal Help

```
Samples: 8K of event 'cycles:u', 4000 Hz, Event count (approx.): 2838883692
Overhead  Shared Object             Symbol
  55.10%  postgres                  [.] KnownAssignedXidsGetAndSetXmin
   1.77%  postgres                  [.] base_yyparse
   1.36%  postgres                  [.] _bt_compare
   1.23%  postgres                  [.] AllocSetAlloc
   1.13%  postgres                  [.] hash_search_with_hash_value
```

Отмена запроса



Гарантии синхронной репликации



Гарантии нарушаются

- Хитрой отменой запроса
- Перезапуском primary ноды
- Падением бэкенда

Disable cancellation of executed locally query

ALTER SYSTEM SET synchronous_commit_cancelation to off;

https://commitfest.postgresql.org/31/2402/

Но это только частичное решение

Рестарт мастера по-прежнему делает нереплицированные данные – видимыми. Проблему надо решать на стороне HA-систем.

Больше информации по проблеме

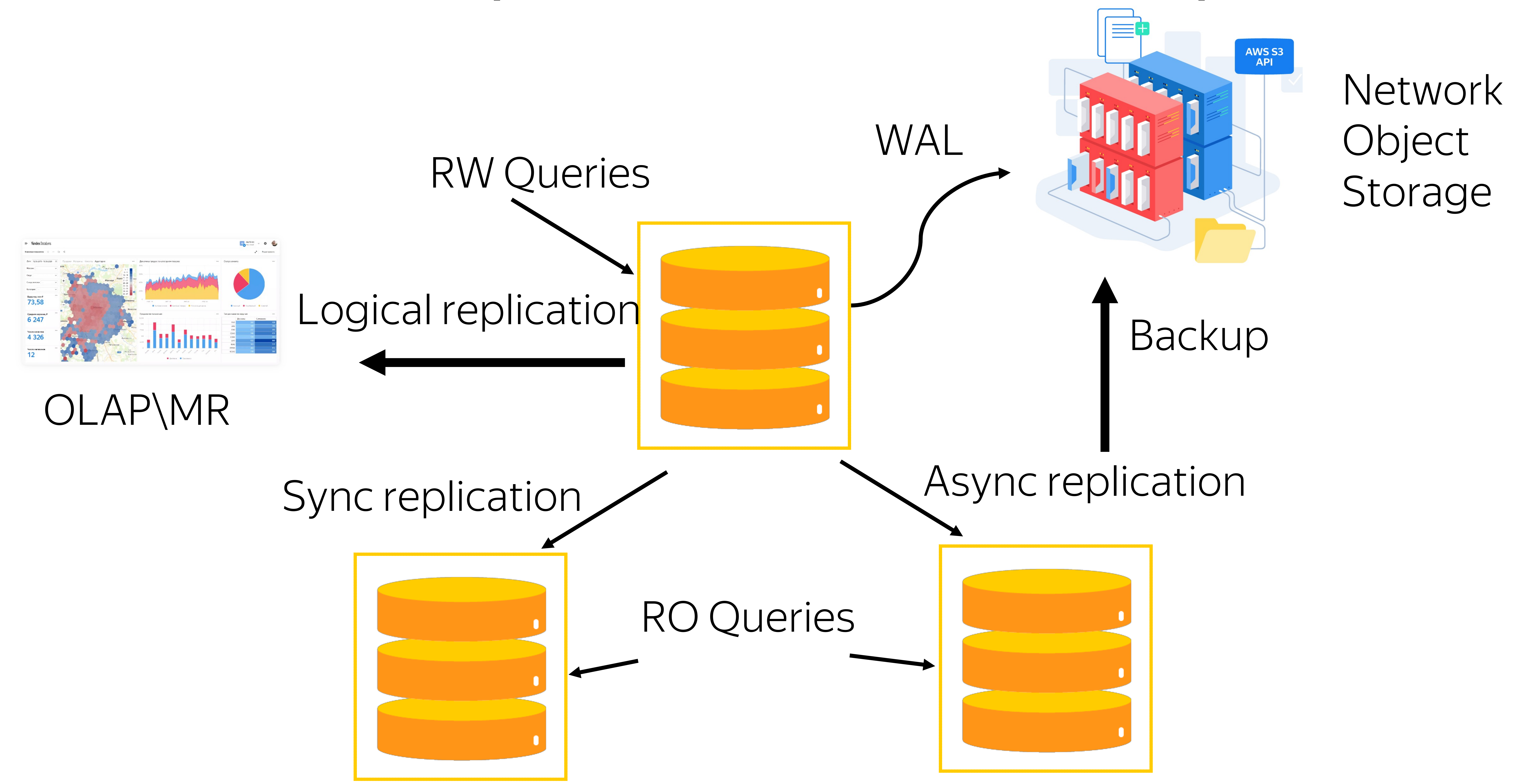
В конспекте PGCon unconference 2020

https://wiki.postgresql.org/wiki/PgCon_2020_Developer_Unconference
/Edge_cases_of_synchronous_replication_in_HA_solutions

Changed data capture



Логическая репликация на мастере



Для логической репликации нужен слот

Логическая репликация запускается с LSN слота.

> **Но слот можно создать только на текущем LSN, не в прошлом.**

Невозможно создать слот в том же LSN, где произошёл failover.

Hack the PostgreSQL

Мы просто создали расширение pg_tm_aux, которое создаёт слот в прошлом

- Есть риск catalog vacuum после переключения мастера
- Для PG 10,11,12,13,14 других вариантов уже точно не будет
- Но есть надежда на светлое будущее

Synchronous standby names

Логическая репликация может быть впереди

- Синхронной реплики
- Кворума

Возможно стоит создать post_synchronous_standby_names?

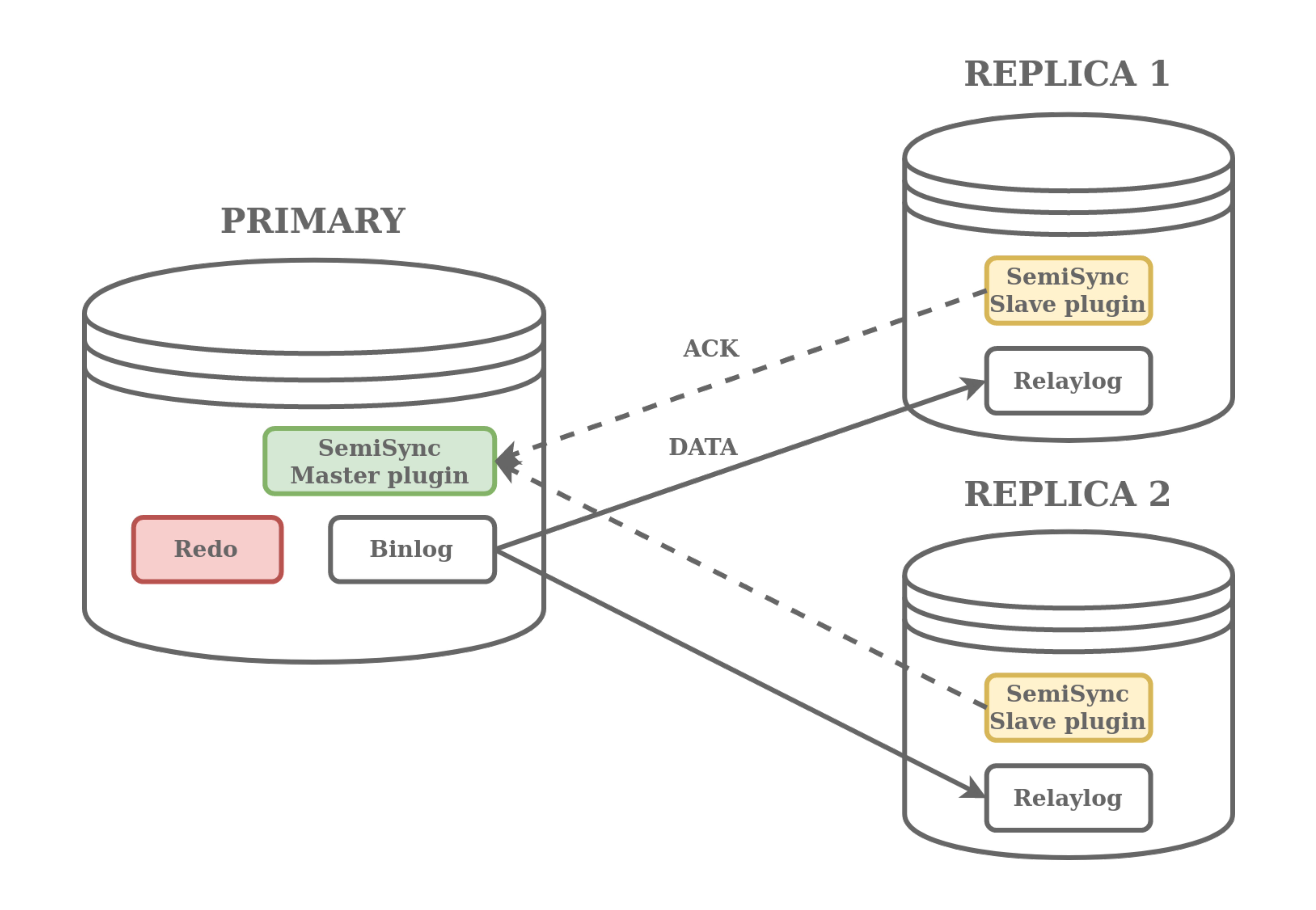
MySQL

Модель консистентности

Read Your Writes

- Использование синхронной / кворумной репликации
- Запись и чтение с мастера

Синхронная репликация в MySQL



Настройка синхронной репликации

Мастер

- plugin_load_add = 'rpl_semi_sync_master=semisync_master.so'
- rpl_semi_sync_master_enabled = 1
- rpl_semi_sync_master_timeout = 31536000000
- rpl_semi_sync_master_wait_for_slave_count = 1

Реплика

- plugin_load_add = 'rpl_semi_sync_slave=semisync_slave.so'
- rpl_semi_sync_slave_enabled = 1

Failover

Что должна сделать HA-утилита ?

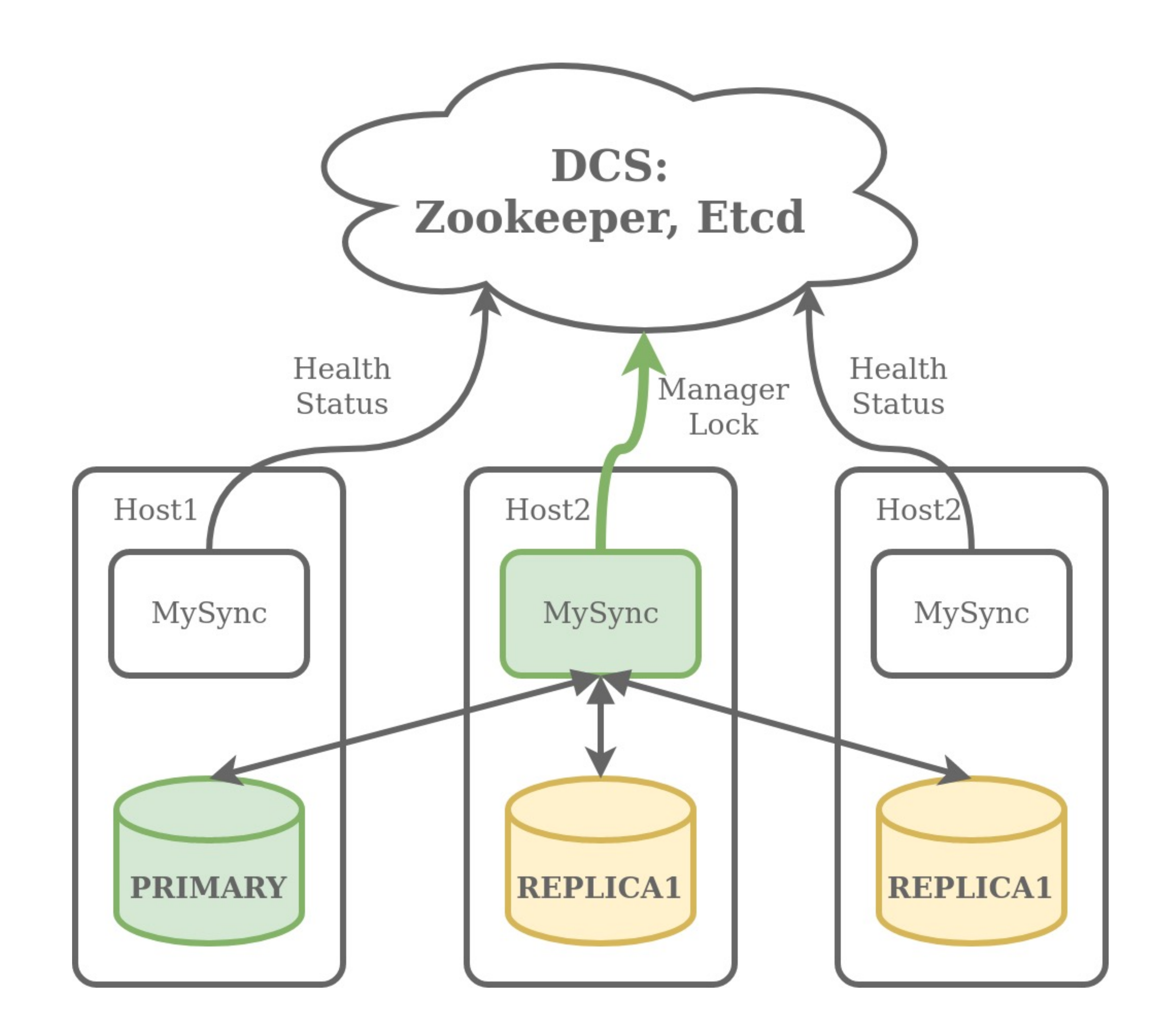
- Определить отказ мастера
- Надежно закрыть старый мастер
- Выбрать лучшую реплику
- Повернуть все реплики на нее
- Открыть новый мастер



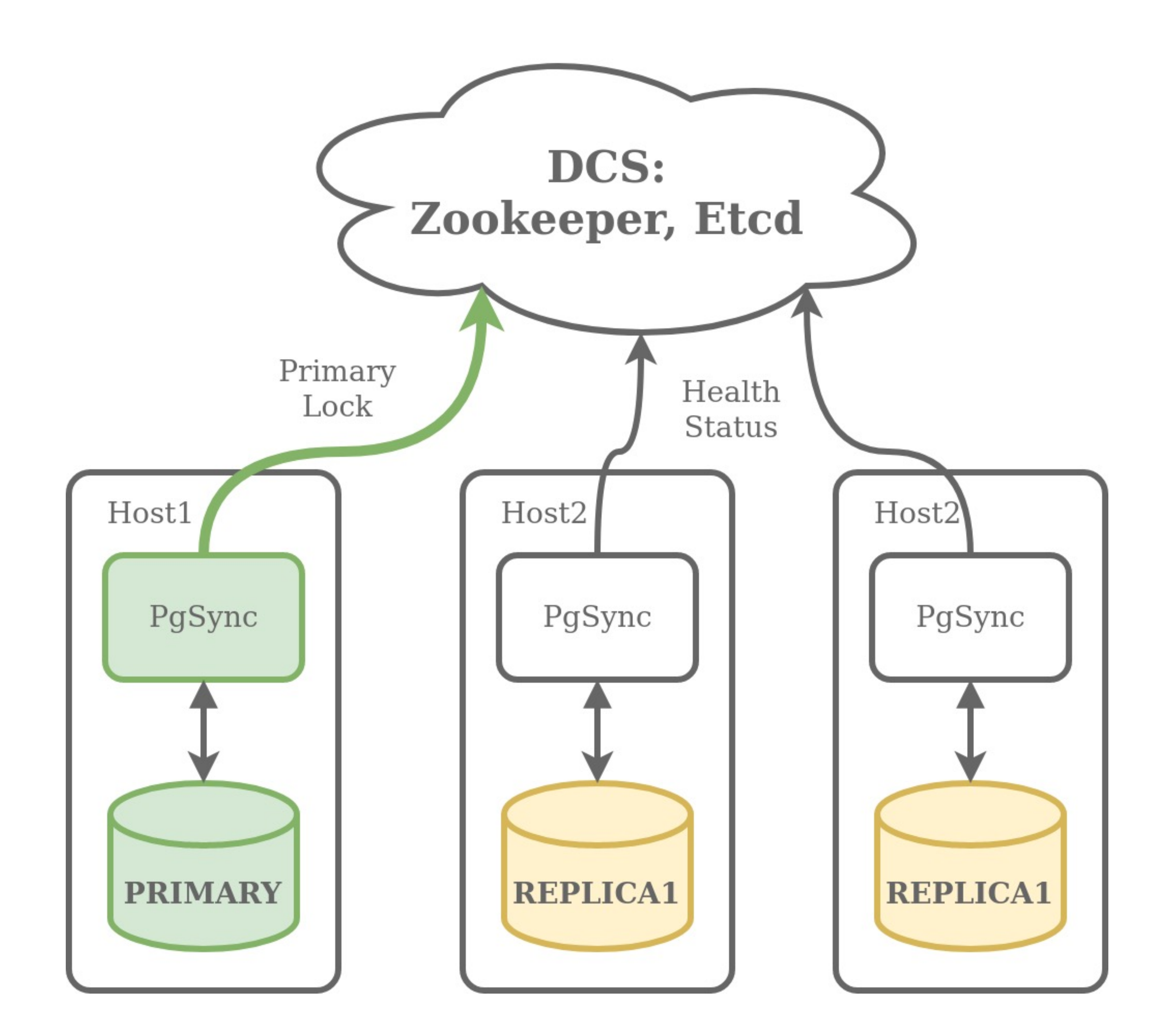
Проблема 0: кто у руля ?



Решение (MySQL)



Решение (Postgres)



Проблема 1: Dead, deader, deadest

Что считать отказом мастера?

- Отказ железа
- Отказ сети
- OOM, crash
- Перегрузка

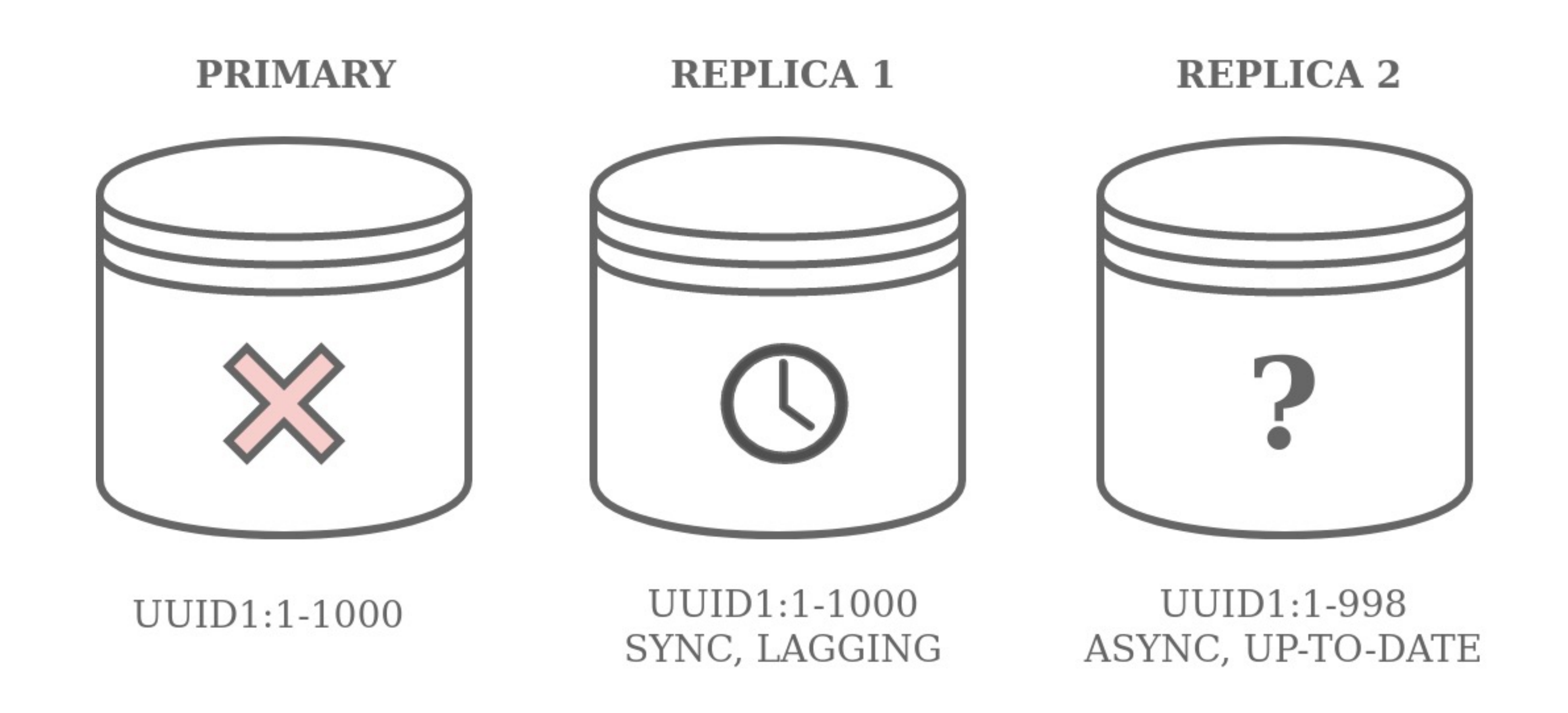


Решение

Отказ мастера – любая ошибка при SELECT 1, но...

- Проверить дважды с разных хостов
- А работает ли репликация ?
- DELAY - проблема повторяется в течение минуты
- COOLDOWN - не делаем failover слишком часто

Проблема 2: replication lag



Решение

- Выбираем реплику, на которой с max (Executed + Retrieved) GTIDset

- Как уменьшить replication_lag ?
  - Правильные индексы в таблицах
  - Небольшие транзакции
  - pt-online-schema-change вместо ALTER

Если доступа к приложению нет ?

Настройки MySQL

- slave_rows_search_algorithms = INDEX_SCAN,HASH_SCAN
- slave_parallel_type = LOGICAL_CLOCK
- slave_parallel_workers = 8
- innodb_flush_log_at_trx_commit = 2 ???

Проблема 3: потеря сети в процессе COMMIT

```
mysql> SHOW PROCESSLIST;

*************************** 4. row ***************************
Id: 39246
User: admin
Command: Killed
Time: 256
State: Waiting for semi-sync ACK from slave
Info: commit

*************************** 8. row ***************************
Id: 39357
User: admin
Command: Query
Time: 122
State: Waiting for commit lock
Info: set global read_only = ON
```

Решение

- Рестарт сервера
- Отключение semisync плагина
  - SET GLOBAL rpl_semi_sync_master_enabled = 0

Смотри метод MYSQL_BIN_LOG::ordered_commit

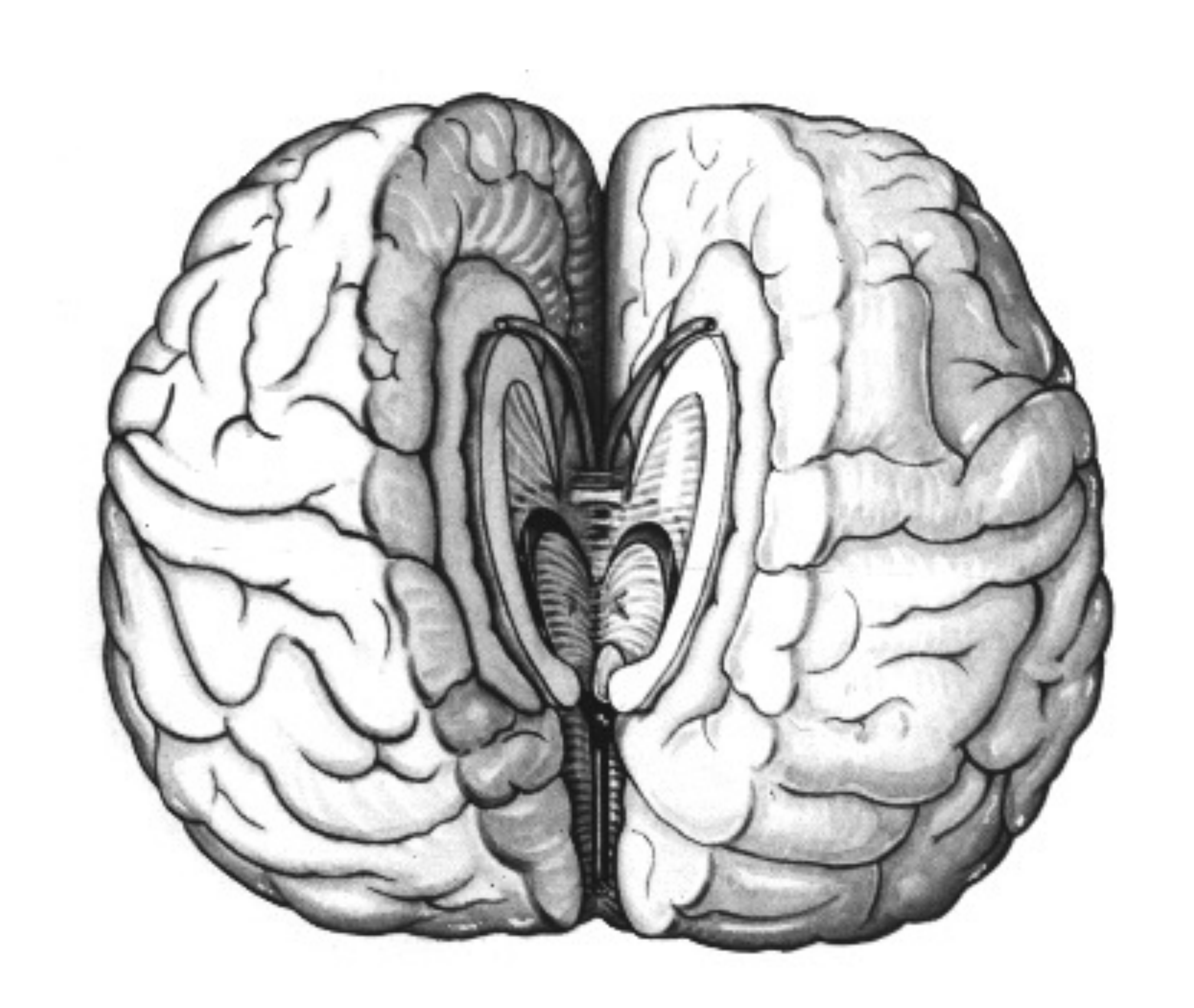
Проблема 4: лишний commit

Старый мастер

- UUID1:1-1003 <= ooops...

Новый мастер

- UUID1:1-1000,UUID2:1-33



Решение

Сервер при старте должен быть невидим для приложения

- read_only = ON
- super_read_only = ON
- offline_mode = ON

Если splitbrain действительно случился..

- rewind ? gh-mysql-rewind, mariadb-binlog --flashback
- переналивка

Что еще требуется от НА-утилиты ?

- switchover
- Управление not-HA-репликами
- Слежение за свободным местом
- Закрытие отставших реплик



За рамками

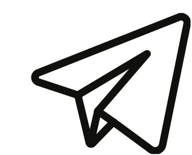
Высокая доступность – не только failover

- Балансировка запросов / service discovery
- Переналивка упавших / отставших хостов
- Резервное копирование

Ждём вопросы

Андрей Бородин

 x4mmm @yandex-team.ru

 x4mmm

Дмитрий Смаль

 mialinx@yandex-team.ru

mialinx